



3.5 Two loops at once

Control · Robot club · about 20 min

Name _____

Class _____

Date _____

What this lesson is about

Heading and sideways controllers together on a holonomic robot.

- Heading alone is not enough
- Add a sideways controller
- Tune them separately
- Holding a line that is not $x = 0$

Questions

6 marks in all

1. The heading is held perfectly, but the robot still ends up to the left. Why? [1 mark]
 - ☐ A It slides sideways even while pointing straight
 - ☐ B The heading gain is too low
 - ☐ C `wrapped()` has the wrong sign
 - ☐ D `position()` cannot measure x
2. In `drive(70, x_error * 10, heading_error * 4)`, what does each number do? [1 mark]
 - ☐ A 70 drives forward, the sideways part holds x , the rotation holds the heading
 - ☐ B 70 holds the heading, the second holds speed, the third holds x
 - ☐ C All three hold the heading
 - ☐ D The first two hold x and the third is the speed
3. `target_x` is 10, `x` is 4, and the sideways gain is 10. What sideways value does `(target_x - x) * 10` give? [1 mark]

4. What does this program print? [1 mark]

```
def wrapped(h):  
    return (h + 180) % 360 - 180  
  
target_x = 10  
x = 13  
heading = 355  
print("sideways", (target_x - x) * 10)  
print("rotation", wrapped(0 - heading) * 4)
```

5. Why do the sideways and heading controllers not fight each other? [1 mark]
 - ☐ A On this robot, rotation and sideways motion are separate
 - ☐ B They take turns, one per loop
 - ☐ C The gains are the same
 - ☐ D They do fight, which is why the robot wobbles

6. What is the right way to tune two gains?

[1 mark]

- ☐ A Change one at a time and watch what it does
- ☐ B Change both together by the same amount
- ☐ C Set both as high as they will go
- ☐ D Copy the gain from lesson 3.3

The task: on the rails

The zone is only 6 cm wide. Finish inside it, facing 0 within 5 degrees, on the curving, leaking robot.

```
# the two lines every program starts with: the commands, then the robot
from bugbot import *
connect()

def wrapped(h):
    return (h + 180) % 360 - 180

while position()[1] < 65:
    error = wrapped(0 - heading())
    # forward, sideways, rotation: -100 to 100 each, until the next command
    drive(70, 0, error * 4)
    # pause 0.1 s (the robot keeps doing what it was told)
    wait(0.1)
# all motors off
stop()
```

Plan your program here, then type it in and press Run.



Do it on the robot

www.bugbotlab.com/learn/3-5-two-loops-at-once/

The simulator checks it and tells you when it passes. Nothing to install, no account.

Challenges

1. Add a third controller: forward speed from the distance to the end of the zone, so it parks at $y = 75$ exactly.
2. Follow the line $x = -10$, then $x = 10$, switching halfway.
3. Write `hold_line(target_x, until_y)` as a function. You will want it in lesson 3.6.